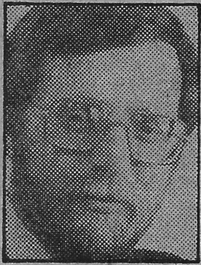

EXECUTIVE COMPUTING

HILLEL SEGAL

Profits tied to data base organization

For many businesses, their most valuable asset is not buildings or equipment, inventory, patents, or products. It is their **computerized data**, such as client records, orders and accounts receivable.

The way the data is organized on disk, and how it is used, often determines how many people are needed for the input and output tasks, and how big the disk files need to be. This dictates the required processing speed of the computer and the company's investment in data processing.

In other words, the mundane question of data-base design has broad ramifications that affect the way the business is conducted — and ultimately its profitability.

Therefore, to get the most from your computers, it's important to understand something called "relational data-base design" The technique is well known in the mainframe world, but not in the small-computer arena.

This week's column describes the basic theory and advantages. Next week, I will review the most popular relational data-base programs: dBASE III, R:base 5000 and PowerBase; along with the non-relational best-seller pfs:File.

Non-relational disadvantages

The underlying tenet of relational data-base design is that it is wasteful to repeat the same information — over and over again — throughout the data base.

Let's say you have an order-entry system that allows your clerk to take orders over the telephone and enter them into the computer. The computer then produces a shipping document and an invoice to be sent to the customer.

Using a simple, non-relational data base program, the clerk is given a form to fill-in on the computer screen for each new order. The customer's name, billing address and shipping address is typed in first. Then each item being dress is typed in. Then each item being ordered is typed in, along with the unit prices. The program automatically does the arithmetic, adds appropriate shipping charges and taxes, and calculates the total — all while the customer is still on the telephone.

Once a day, all the shipping documents and invoices are printed out. This is a common and productive application for a small computer.

But if you looked at the computer's files, you'd see that an enormous amount of information is needlessly duplicated: product descriptions, unit prices, and all the customer information.

The resulting duplication in the computer could be staggering.

In this example, which is not uncommon, a particular customer's name and address may appear hundreds of times, and the same product descriptions may appear thousands of times during a single year's period.

As crazy as this sounds, it is exactly the way most systems are set up. Because the initial savings of computerization may still be significant over manual systems, such systems tend to grow to unmanageable size before they are discovered.

Advantages of relational systems

By now, you've probably guessed that a relational system does away with the duplication. That's the key.

To computerize the above application more efficiently, you first might set up a simple customer file, with all the data that typically does not change from order to order: Billing name and address, shipping name and address, payment terms, etc. Most important, you would assign each one a "customer number."

Next, you might set up a product file, with a short description of each product, exactly the way it should appear on the invoices. This file might also include the current price of each item. Not surprisingly, each product would be assigned a "product number."

Then, with a data-base system that allows you to "relate" the files (thus the name "relational"), you can make your application sing! You can set up a "new order file" that automatically calls the information it needs from the other files based on the customer numbers and product numbers.

As a result, your file containing new orders need only contain customer numbers and product numbers! Whenever a new order is entered on the screen, the data from the related files is filled in for you — so it creates the illusion that all the descriptive data is in each file.

When taking a new order over the telephone, the operator just enters the customer number and the part numbers into the computer. Instantly, all the information from the customer and product files appear on the screen, just as if they were typed in fresh. And the arithmetic is done automatically so the total costs can be verified with the customer.

The bottom line

The bottom line is: Systems set up this way often have files that are just a fraction of the size of non-relational data base systems. Also, they typically run faster, are less prone to error, and are less expensive to maintain.

Hillel Segal is an independent computer consultant and editor of the Executive Computing Newsletter, published by the Association of Computer Users. He can be reached at ACU, P.O. Box 9003, Boulder 80301.